CLAIMS

What is claimed is:

1. In a microprocessor implementing a subtraction division algorithm for determining a quotient of a floating point divide operation by iteratively subtracting divisor multiples from a partial remainder, each subtraction producing an unshifted partial remainder in carry-save form that is multiplied by the radix of the algorithm to obtain a next partial remainder, a method of determining quotient digits comprising:

calculating a carry-propagate form of a plurality of most significant bits to the right of the radix point of the unshifted partial remainder; and

assigning a next quotient digit to be a whole number value of at least one of the most significant bits to the right of the radix point of the unshifted partial remainder.

2. The method of determining quotient digits as defined in claim 1 wherein calculating a carry-propagate form of a plurality of most significant bits to the right of the radix point of the unshifted partial remainder further comprises calculating the carry-propagate form of one of a number of bits selected from the following table based on a radix of the SRT algorithm:

Radix	Number of bits to convert to carry-propagate form
2	3
4	4
8	5
16	6
32	7

3. The method of determining quotient digits as defined in claim 1 wherein assigning a next quotient digit to be a whole number value of at least one of the most significant bits to the right of

the radix point of the unshifted partial remainder further comprises assigning a next quotient digit to be the whole number value of one of a number of most significant bits selected from the following table based on the radix of the divider algorithm:

Radix	Number of bits representing the next quotient digit
2	1
4	2
8	3
16	4
32	5

4. The method of determining quotient digits as defined in claim 1 wherein calculating a carry-propagate form of a plurality of most significant bits to the right of the radix point of the unshifted partial remainder further comprises calculating the carry-propagate form of four bits to the right of the radix point in a Radix-4 divider algorithm.

5. The method of determining quotient digits as defined in claim 4 wherein assigning a next quotient digit to be a whole number value of at least one of the most significant bits to the right of the radix point of the unshifted partial remainder further comprises assigning the next quotient digit to be a whole number value represented by two most significant bits to the right of the radix point of the unshifted partial remainder in a Radix-4 divider algorithm.

6. The method of operating the iterative cell as defined in claim 1 wherein prescaling the divisor such that a whole number part of the partial remainder is the quotient digit further comprising prescaling the divisor to be in the range:

$$1 \le D < 1 + (1/R)$$

where D is the divisor, and R is the radix of the floating point division logic.

7. The method of operating the iterative cell as defined in claim 1 further comprising assigning a next quotient digit from the set of possible quotient digits

$\{-(R-1), -(R-2), \ldots, -(R-N), -0, 0, (R-N), \ldots (R-2), (R-1), R\}$

where R is the radix and $N=R-1$.

8. The method of operating the iterataive cell as defined in claim 7 further comprising assigning a next quotient digit from the set

{-3,-2,-1,-0,0,1,2,3,4}

for a Radix-4 subtractive division algorithm.

9. A microprocessor having a floating point unit comprising a subtractive division (SD) system that comprises a prescale unit coupled to a divisor and a dividend, and wherein the prescale unit scales the divisor and the dividend, and also calculates divisor multiples, the microprocessor having an SD cell comprising:

a first adder logic having a first set of inputs coupled to the divisor multiples and a second set of inputs coupled to a partial remainder from a previous SD cell, the first adder logic producing bits to the right of the radix part of a resultant in carry-save form and also producing a first carry-out of the resultant;

a second adder logic coupled to at least one of the most significant bits of the resultant, the second adder logic converting the at least one of the most significant bits of the resultant to carry-

propagate form and also producing and a second carry-out of the conversion of the plurality of significant bits of the resultant to carry-propagate form; and

a decode logic coupled to the first carry-out, the second carry-out, and the at least one of the most significant bits of the resultant in carry-propagate form, and wherein the decode logic selects the quotient digit to be the whole number value of the at least one of the most significant bits of the resultant in carry-propagate form.

10. The microprocessor having an SD cell as defined in claim 7 wherein said decode logic selects the quotient digit to be the whole number value of the at least one of the most significant bits of the resultant in carry-propagate form when only one of the first and second carry-outs is asserted.

11. The microprocessor having an SD cell as defined in claim 10 wherein said decode logic selects the quotient digit to be a negative one's compliment of the at least one of the most significant bits of the resultant in carry-save form when neither of the first and second carry-outs is asserted.

12. The microprocessor having an SD cell as defined in claim 11 wherein said decode logic selects the quotient digit to be the value of the radix of the SRT division system when the both the first and second carry-outs are asserted.

13. The microprocessor having an SD cell as defined in claim 12 wherein said second adder logic converts four of the most significant bits of the resultant to carry-propagate form in a Radix-4 SRT division system.

14. The microprocessor having an SD cell as defined in claim 13 wherein said the decode logic selects the quotient digit to be the whole number value of two of the most significant bits of the resultant in carry-propagate form in a Radix-4 system when only one of the first and second carry-outs is asserted.

15. The microprocessor having an SD cell as defined in claim 9 wherein the said decode logic selects the quotient digit from the set

$\{-(R-1), -(R-2), \ldots, -(R-N), -0, 0, (R-N), \ldots, (R-2), (R-1), R\}$

where R is the radix on $N=R-1$.

16. The microprocessor having an SD cell as defined in claim 9 wherein said decode logic selects the quotient digit from the set

{-3,-2,-1,-0,0,1,2,3,4}

in a Radix-4 subtractive division system.

17. In a microprocessor implementing a floating point division algorithm for determining a quotient of a floating point divide operation comprising at least one cell iteratively subtracting divisor multiples from a partial remainder, each subtraction producing a resultant that is multiplied

by the radix of the algorithm to obtain a next partial remainder, a method of operating a cell of the floating point division algorithm comprising:

calculating a carry-save form of the resultant, each digit of the resultant represented by a sum bit and a carry bit, and the calculating the carry-save form of the resultant also producing a first carry-out;

calculating a carry-propagate form of a plurality most significant bits to the right of the radix point of the resultant, and the calculating the carry-propagate form also producing a second carry-out; and

assigning a next quotient digit to be a value directly indicated by at least one of the most significant bits to the right of the radix point of the resultant multiplied by the radix if only one of the first and second carry-outs are asserted.

18. The method of operating a cell of the floating point division algorithm as defined in claim 17 wherein calculating a carry-propagate form of a plurality most significant bits to the right of the radix point of the resultant further comprises calculating four of the most significant bits to the right of the radix point for a Radix-4 divider.

19. The method of operating a cell of the floating point division algorithm as defined in claim 18 wherein assigning a next quotient digit to be a value directly indicated by at least one of the most significant bits to the right of the radix point of the resultant multiplied by the radix further comprises assigning the next quotient digit to be a value directly indicated by two of the most significant bits to the right of the radix point multiplied by the radix.

20. The method of operating a cell of the floating point division algorithm as defined in claim 17 further comprising assigning the next quotient digit to be a negative one's complement of at least one of the most significant bits to the right of the radix point of the resultant multiplied by the radix if neither of the first and second carry-outs are asserted.

21. The method of operating a cell of the floating point division algorithm as defined in claim 20 wherein assigning the next quotient digit to be a negative one's complement of at least one of the most significant bits to the right of the radix point of the resultant further comprises assigning the next quotient digit to be the negative one's complement of two of the most significant bits to the right of the radix point of the resultant multiplied by the radix if neither of the first and second carry outs are asserted.

22. The method of operating a cell of the floating point division algorithm as defined in claim 20 further comprising assigning the next quotient digit to be a radix value if both the first and second carry outs are asserted.

23. The method of operating a cell of the floating point division algorithm as defined in claim 22 wherein assigning the next quotient digit to be a radix value if both the first and second carry outs are asserted further comprises assigning a next quotient digit to have a value of four in a Radix-4 divider.

24. In a microprocessor having a floating point division logic, a method of operating an iterative cell of the floating point division logic comprising:

subtracting a divisor multiple from a partial remainder from a previous iterative cell to form a resultant;

multiplying the resultant by a radix of the floating point division logic to obtain a partial remainder for the next iterative cell; and

prescaling the divisor such that a whole number part of the partial remainder for the next iterative cell indicates a next quotient digit directly.

25. The method of operating an iterative cell of the floating point division logic as defined in claim 24 wherein prescaling the divisor such that a whole number part the partial remainder for the next iterative cell indicates the next quotient digit directly further comprises prescaling the divisor to be within the range:

$$1 \leq D < 1 + (1/R)$$

where D is the divisor, and R is the radix of the floating point division logic.

26. The method of operating an iterative cell of the floating point division logic as defined in claim 24 wherein subtracting a divisor multiple from a partial remainder from a previous iterative cell to form a resultant further comprises:

adding a two's complement form of the divisor multiple to a carry-save form of the partial remainder from the previous stage to obtain the resultant in carry-save form; and

converting a plurality of bits to the right of the radix point of the resultant to carry-propagate form.

27. The method of operating an iterative cell of the floating point division logic as defined in claim 26 wherein converting a plurality of bits to the right of the radix point of the resultant to carry-propagate form further comprises converting to carry-propagate form of a number of bits selected from the following table based on the radix of the floating point division algorithm:

Radix	Number of bits to convert to carry-propagate form
2	3
4	4
8	5
16	6
32	7

28. A method of selecting a quotient digit in a cell of a Sweeny-Robertson-Toucher (SRT) division system in a microprocessor, the method comprising:

adding at least one most significant bit in carry-propagate form to the right of the radix of a partial remainder from a previous SRT cell to at least one most significant bit to the right of the radix point of a divisor multiple to obtain a first one-hot encoded output; and

increasing a value indicated by the first one-hot encoded output if the addition of lower order bits of the partial remainder from the previous SRT cell and lower order bits of the divisor multiple produces a carry, the increasing producing a second one-hot encoded output, and wherein the second one-hot encoded output directly indicates a quotient digit from a set of possible quotient digits.

29. The method of selecting a quotient digit in a cell of a SRT division system as defined in claim 28 further comprising:

decoding, prior to the adding step, a plurality of lower order bits in carry-save form of the partial remainder from the previous SRT cell to produce a third one-hot encoded output;

summing the third one-hot encoded output with a plurality of lower order bits of the divisor multiple substantially simultaneously with the adding step, the summing producing a shift output; and

wherein the increasing step is based on the shift output of the summing step.

30. The method of selecting a quotient digit in a cell of a SRT division system as defined in claim 29 wherein the adding step further comprises adding two most significant bits to the right of the radix of the partial remainder from a previous SRT cell to two most significant bits to the right of the radix point of a divisor multiple to obtain the first one-hot encoded output in a Radix-4 system.

31. The method of selecting a quotient digit in a cell of a SRT division system as defined in claim 30 wherein the decoding step further comprises decoding three lower order bits in carry-save form of the partial remainder from the previous SRT to produce the third one-hot encoded output in a Radix-4 system.

32. The method of selecting a quotient digit in a cell of a SRT division system as defined in claim 31 wherein the summing step further comprises summing the third one-hot encoded output to three lower order bits of the divisor multiple substantially simultaneously with the adding step.